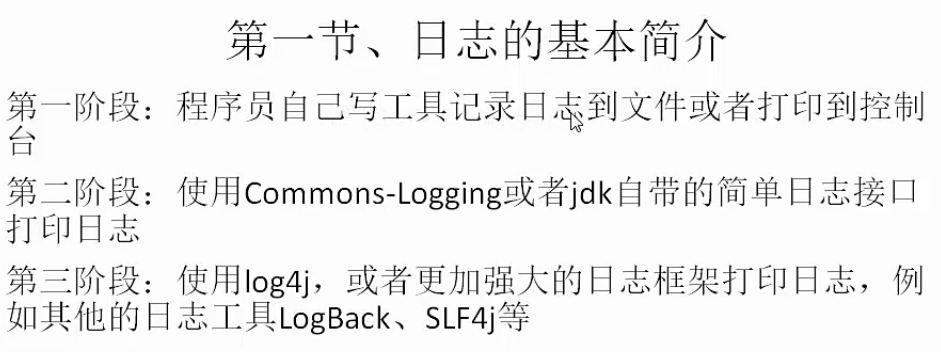
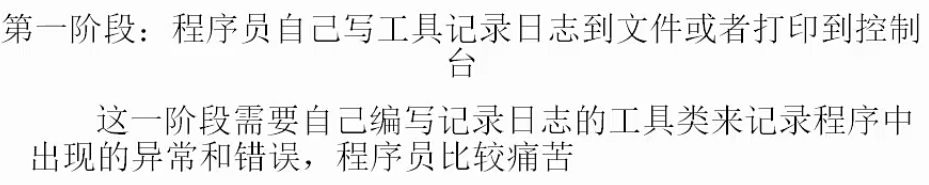
日志基本简介





--------------------------------------------------------------------



第一阶段就是System.out.println() + IO流

---------------------------------------------------------------------

第二阶段 使用commons-logging或者jdk自带的简单日志接口



总结：

(1). Commons-logging和jdk自带的日志功能比较弱 不能详细记录出现的异常或者错误 比如异常出现在哪一行 之类的

(2). Commons-logging通常以**日志接口**的形式体现出来

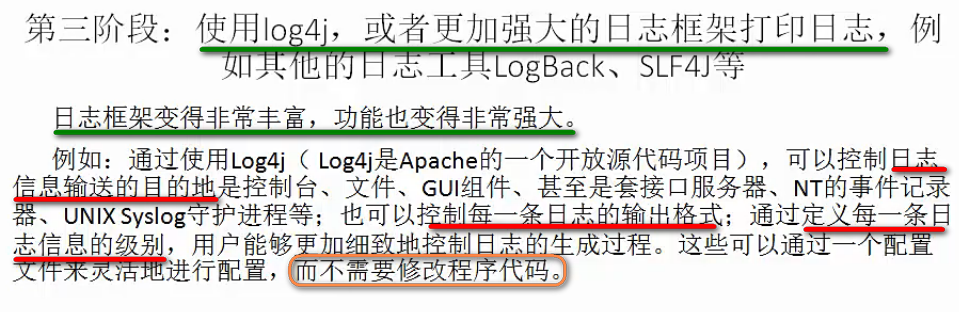
---- commons-logging自带一个接口的实现类SimpleLogger 功能非常弱

(3). Commons-logging通常都是和log4j配合使用

Commons-logging是接口规范 log4j是接口规范的实现类

Commons-logging的好处：修改第三方日志非常容易

第三阶段：使用第三方日志实现的库



Log4j可以把日志输出到 控制台 文件 GUI组件 套接口服务器 NT事件记录器 UnixSyslog守护进程等

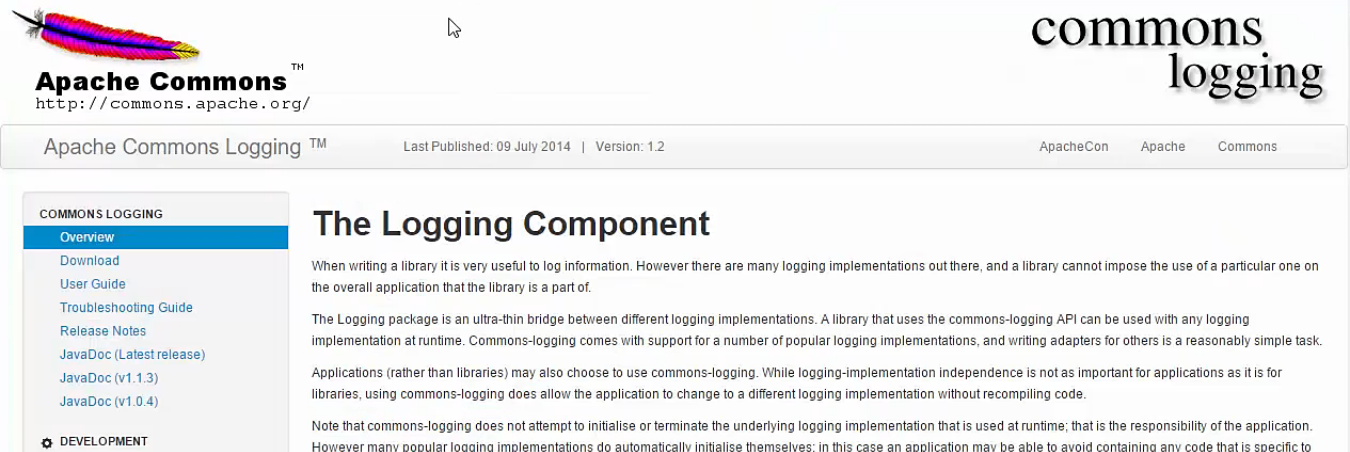
第三方实现由 Logback slf4j等等

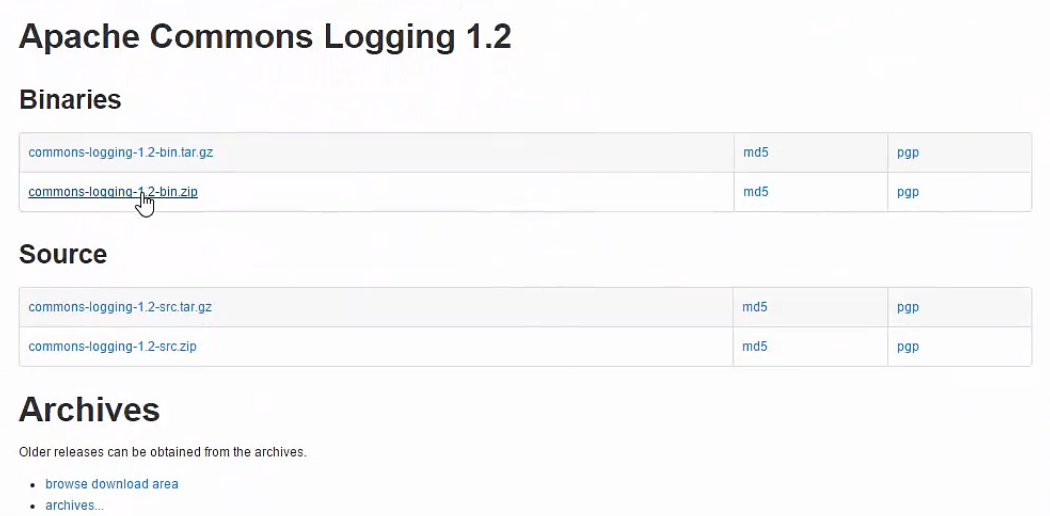
日志框架commons-logging和Log4j应用



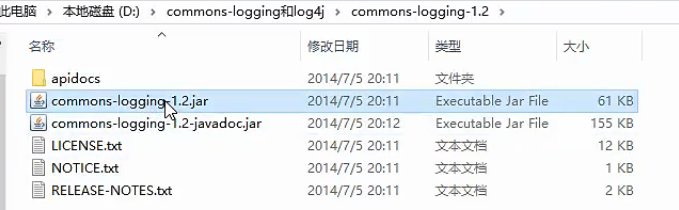








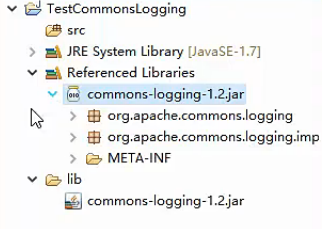
Bin.zip和src.zip都要下载下来

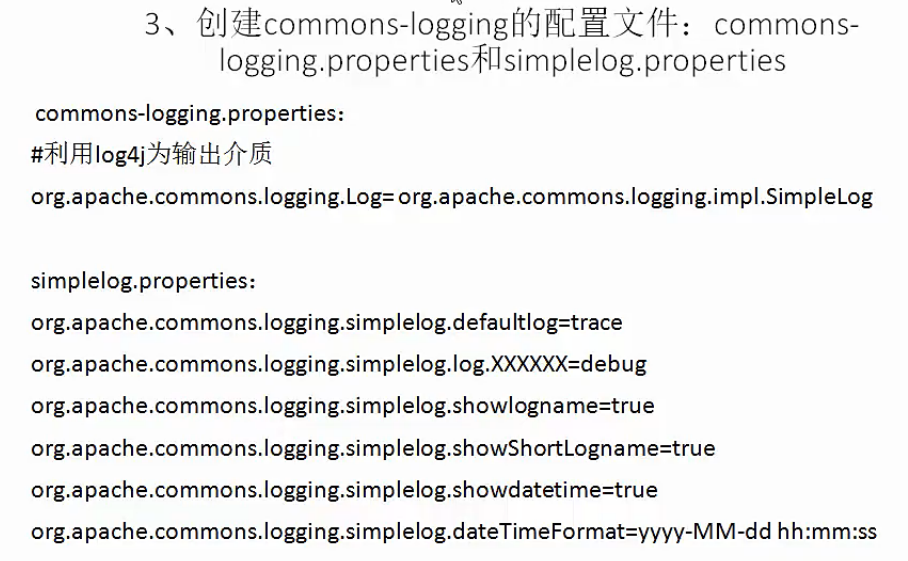


这里面是对应的介绍



在eclipse中进行添加





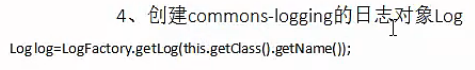
进行配置

这里面首先使用commons-logging自带的实现输出 SimpleLogger



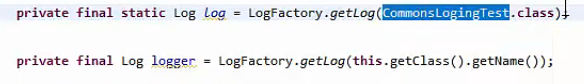
这个就是告诉 我的接口 org.apache.commons.logging.Log这个接口的实现类就是org.apache.commons.logging.impl.SimpleLog

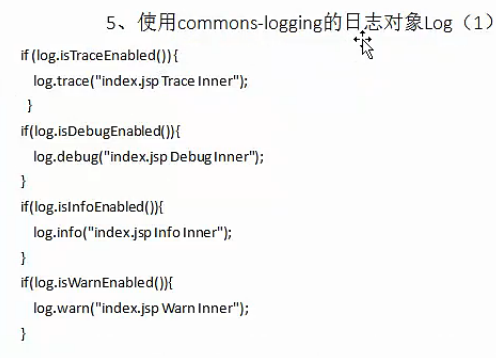
这里面进行简单的配置

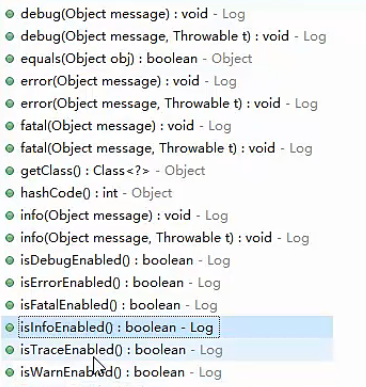




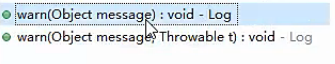
如果不用class可以使用字符串

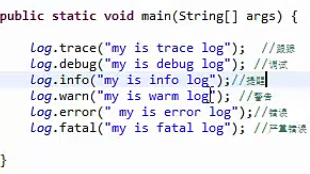




 --- 常用的方法

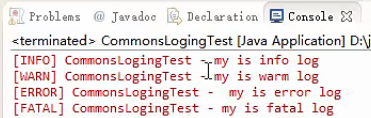
trace() 🡪 跟踪 debug() –> 调试 info() 🡪 提示 warn() 🡪 警告 error() 🡪错误 fatal()->严重的错误（就是致命的）

 每一种都由两种重载的形式 一个直接给出信息 另一种不仅给出要打印的信息 还可以填写异常类



这样有6个级别的信息日志 从trace到fatal级别是从低到高

运行一下



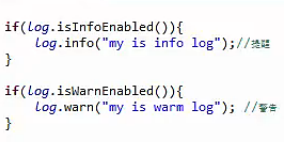
这样就打印出日志

只打出来四个级别的 trace和debug都没有打印出来

默认是trace和debug都不打印的

如何使用呢？

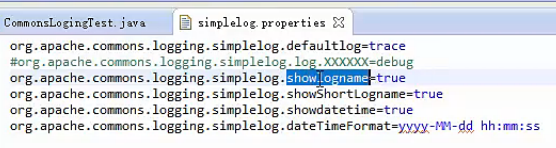
对每一个进行判断 符合条件才进行打印



这样通过配置文件来控制到底哪些日志要打印

【默认仅仅打印INFO级别以上的级别】

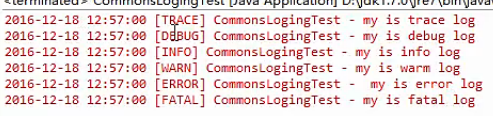
重新配置别的信息simplelog.properties



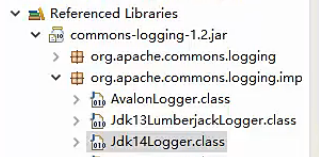
注意 配置的时候 如果value后面有空格 那么 解析出来的true就变成了false 就打印不出来日志了

默认打印的级别是trace以上的日志都进行打印

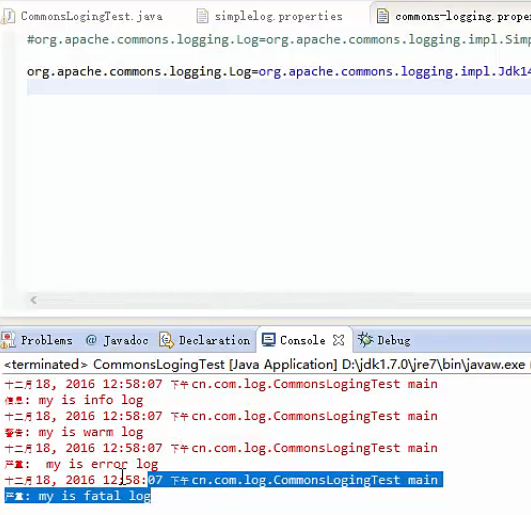
是否显示log名称



这样日志是非常详细的信息

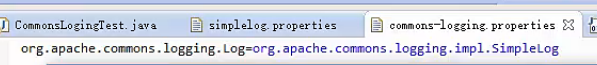


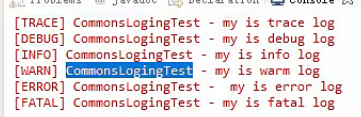
还提供了一个

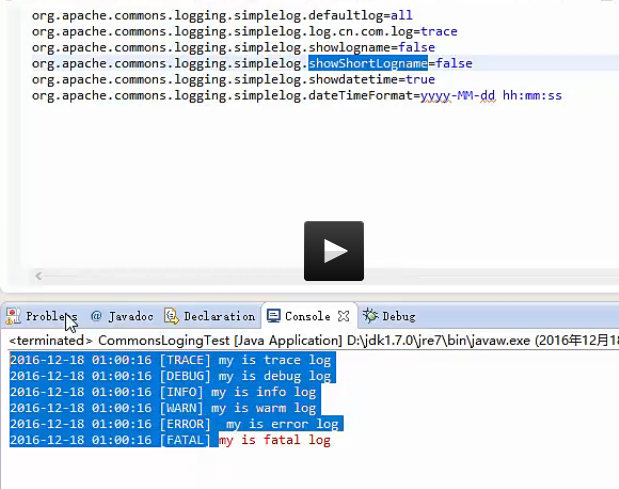
 这个是JDK1.4的日志输出格式

修改一下配置

日期显示改为false







ShowSortLogname 就是不显示类名了